**Student Performance Predictor**

**Team Members**

* Hasnain (Roll No: 2212252)
* Sadam Hussain (Roll No: 2212266)
* Omesh (Roll No: 2212262)

**Project Overview**

The Student Performance Predictor is a web-based application designed to forecast student academic performance based on multiple influencing factors. The system leverages machine learning to analyze various student-related inputs and provides a predictive output that helps educators and students identify potential outcomes. This predictive insight can be used to improve study habits, focus on weaker areas, and tailor educational strategies.

**Objectives**

* To develop a reliable machine learning model capable of predicting student performance using key academic and behavioral indicators.
* To create an interactive web interface that allows users to input relevant student data and receive immediate predictions.
* To demonstrate the application of data science and machine learning techniques within a practical educational context.

**Data and Features**

The model is trained on historical data encompassing the following features:

* **Attendance (%)**: Represents the percentage of classes attended by the student.
* **Previous Grade**: The student’s grade in previous assessments or semesters.
* **Study Hours**: Average daily study time invested by the student.
* **Extracurricular Activities**: Binary indicator (0 or 1) representing involvement in extracurriculars.
* **Assignment Completion (%)**: Percentage of assignments submitted on time.

These features were selected based on their strong correlation with student success and are used as input variables to the predictive model.

**Machine Learning Model**

* **Algorithm Used**: Random Forest Classifier, chosen for its robustness and ability to handle both categorical and numerical data.
* **Model Training**: The model was trained on labeled student performance data, fine-tuned, and validated to ensure accuracy and reliability.
* **Model Persistence**: The trained model was serialized and saved using Joblib, enabling seamless integration with the Flask web application for runtime prediction.

**Web Application Development**

* **Framework**: Flask was used for building the backend web server due to its simplicity and flexibility.
* **Frontend**: HTML and CSS provide a clean and user-friendly interface for data input and display of prediction results.
* **Integration**: User input is captured via HTML forms and sent to Flask, which passes the data to the machine learning model to compute predictions. The results are then dynamically rendered on the web page.

**Technical Implementation Details**

* The Flask app loads the serialized Random Forest model from the model directory at startup.
* User inputs are validated and transformed into a feature vector, which is passed to the model’s predict() method.
* Static files such as CSS are served correctly via Flask’s static folder configuration.
* The template rendering uses Jinja2 to dynamically display prediction outcomes on the frontend.

**Challenges and Solutions**

* **File Path Issues**: Initially, incorrect relative paths caused the model file to not be found. This was resolved by ensuring the app runs from the project root and adjusting paths accordingly.
* **Serving Static Files**: CSS was not loading due to improper linking; fixed by using Flask’s url\_for('static', filename='style.css') method within the HTML template.
* **Feature Name Warning**: The model generated warnings about feature names; addressed by carefully matching input data structure during prediction.

**Conclusion**

The Student Performance Predictor effectively demonstrates the practical application of machine learning in the educational domain. By integrating a trained predictive model with a simple Flask web app, the project offers an accessible tool for anticipating student outcomes based on measurable inputs. This solution highlights the potential of data-driven approaches to support student success and can be further enhanced with larger datasets, more features, and advanced modeling techniques.